

Refine Search

Search Results -

Term	Documents
(11 AND 3).USPT.	0
(L3 AND L11).USPT.	0

Database:

US Pre-Grant Publication Full-Text Database
 US Patents Full-Text Database
 US OCR Full-Text Database
 EPO Abstracts Database
 JPO Abstracts Database
 Derwent World Patents Index
 IBM Technical Disclosure Bulletins

Search:

L13

Refine Search

Recall Text

Clear

Interrupt

Search History

 DATE: Thursday, April 07, 2005 [Printable Copy](#) [Create Case](#)
Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=YES; OP=ADJ

<u>L13</u>	13 and L11	0	<u>L13</u>
<u>L12</u>	13 and L10	2	<u>L12</u>
<u>L11</u>	12 and L10	6	<u>L11</u>
<u>L10</u>	11 and L9	20	<u>L10</u>
<u>L9</u>	700/1,100,221.ccls.	553	<u>L9</u>
<u>L8</u>	11 and L7	7	<u>L8</u>
<u>L7</u>	718/102.ccls.	721	<u>L7</u>
<u>L6</u>	11 and 12 and 13 and L5	11	<u>L6</u>
<u>L5</u>	identifier\$1	54726	<u>L5</u>
<u>L4</u>	idnetifier\$1	0	<u>L4</u>
<u>L3</u>	task\$1 near2 order	3338	<u>L3</u>
<u>L2</u>	control\$ near1 program\$1	70859	<u>L2</u>
<u>L1</u>	process sequenc\$ or (sequenc near3 process\$)	10062	<u>L1</u>



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used [process](#) [sequence](#) [Scheduling](#) [control](#) [program](#)

Found 102,219 of 153,034

Sort results
by
[Save results to a Binder](#)Try an [Advanced Search](#)Try this search in [The ACM Guide](#)Display
results
[Search Tips](#)
☐ Open results in a new
window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**1** [Proper termination of flow-of-control in programs involving concurrent processes](#)

Kim Gostelow, Vincent G. Cerf, Gerald Estrin, Saul Volansky

August 1972 **Proceedings of the ACM annual conference - Volume 2**Full text available:  [pdf\(743.31 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents new results from a study of flow-of-control in programs involving concurrent processes. Within the framework of flow-of-control, this paper: 1) defines a property of parallel program behavior called proper termination (PT); 2) shows that any properly terminating program containing a specification of the resources it requires will be free of deadlock; 3) identifies other important features of PT programs; and < ...


Keywords: Correctness, Deadlock detection, Parallel program schemata, Resource allocation

2 [Parallel execution of prolog programs: a survey](#)

Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo

July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,

Volume 23 Issue 4

Full text available:  [pdf\(1.95 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Since the early days of logic programming, researchers in the field realized the potential for exploitation of parallelism present in the execution of logic programs. Their high-level nature, the presence of nondeterminism, and their referential transparency, among other characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. At the same time, the fact that the typical applications of logic programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, prolog

3 [The state of the art: Proper termination of flow-of-control in programs involving concurrent processes](#)


Kim Gostelow, Vincent G. Cerf, Gerald Estrin, Saul Volansky


November 1972 **ACM SIGPLAN Notices**, Volume 7 Issue 11

Full text available:  [pdf\(697.55 KB\)](#) Additional Information: [full citation](#), [references](#)

4 Computing curricula 2001

September 2001 **Journal on Educational Resources in Computing (JERIC)**

Full text available:  [pdf\(613.63 KB\)](#)

 [html\(2.78 KB\)](#)


Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



5 Deterministic Processor Scheduling

Mario J. Gonzalez

September 1977 **ACM Computing Surveys (CSUR)**, Volume 9 Issue 3

Full text available:  [pdf\(2.86 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



6 Continuous program optimization: A case study

Thomas Kistler, Michael Franz

July 2003 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,
Volume 25 Issue 4

Full text available:  [pdf\(877.67 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#),
[review](#)

Much of the software in everyday operation is not making optimal use of the hardware on which it actually runs. Among the reasons for this discrepancy are hardware/software mismatches, modularization overheads introduced by software engineering considerations, and the inability of systems to adapt to users' behaviors. A solution to these problems is to delay code generation until load time. This is the earliest point at which a piece of software can be fine-tuned to the actual capabilities of the ...


Keywords: Dynamic code generation, continuous program optimization, dynamic reoptimization



7 A tool for the deterministic scheduling of real-time programs implemented as periodic Ada tasks

E. W. Giering, T. P. Baker

September 1994 **ACM SIGAda Ada Letters , Proceedings of the second international symposium on Environments and tools for Ada**, Volume XIV Issue SI

Full text available:  [pdf\(1.57 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

In this paper, we describe an experimental tool for the scheduling and execution of real-time programs on a single processor. This tool accepts a real-time program implemented as a system of periodic tasks written in a subset of Ada. It translates the program into equivalent Ada source code in which the task bodies are executed by a run-time dispatcher according to a deterministic, cyclic schedule. The schedule is represented as a table of scheduling actions describing the execution of the progra ...



8 A scheduling philosophy for multi-processing systems

Butler W. Lampson

January 1967 **Proceedings of the first ACM symposium on Operating System Principles**

Full text available:  [pdf\(1.51 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

One of the essential parts of any computer system is a mechanism for allocating the processors of the system among the various competitors for their services. These allocations





US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used process sequence control program

Found 101,769 of 153,034

Sort results
by[Save results to a Binder](#)[Try an Advanced Search](#)[Try this search in The ACM Guide](#)Display
results[Search Tips](#)☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**1** [Digital control of industrial processes](#)

Cecil L. Smith

September 1970 **ACM Computing Surveys (CSUR)**, Volume 2 Issue 3Full text available: [pdf\(2.11 MB\)](#)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**2** [The hierarchical simulation language HSL: a versatile tool for process-oriented simulation](#)

D. P. Sanderson, R. Sharma, R. Rozin, S. Treu

April 1991 **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, Volume 1 Issue 2Full text available: [pdf\(2.68 MB\)](#)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)**Keywords:** C++, HSL, hierarchy, inheritance, interpreter, modularity, process, simulation programming language**3** [Survey paper: A survey of control structures in programming languages](#)

David A. Fisher

November 1972 **ACM SIGPLAN Notices**, Volume 7 Issue 11Full text available: [pdf\(962.30 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The control structure of programming languages and their development are examined. Languages studied range from machine and assembly languages to procedure and problem-oriented languages. The emphasis, however, is on the control structures themselves, whether in current languages or proposed. Both implicit global interpretation rules for programming languages and explicit control operations are discussed. Many control structures developed through specialization from a small set of primitive sequ ...

4 [Shared memory consistency conditions for non-sequential execution: definitions and programming strategies](#)


Hagit Attiya, Soma Chaudhuri, Roy Friedman, Jennifer L. Welch

August 1993 **Proceedings of the fifth annual ACM symposium on Parallel algorithms and architectures**

Full text available:  pdf(1.14 MB)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

5 Analyzing the communication topology of concurrent programs

Christopher Colby

June 1995 **Proceedings of the 1995 ACM SIGPLAN symposium on Partial evaluation and semantics-based program manipulation**Full text available:  pdf(1.11 MB)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

6 Production of large computer programs

H. D. Benington

March 1987 **Proceedings of the 9th international conference on Software Engineering**Full text available:  pdf(1.24 MB)Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

The paper is adapted from a presentation at a symposium on advanced programming methods for digital computers sponsored by the Navy Mathematical Computing Advisory Panel and the Office of Naval Research in June 1956. The author describes the techniques used to produce the programs for the Semi-Automatic Ground Environment (SAGE) system.

7 An experimental investigation of the effect of program structure on program understanding

Tom Love

March 1977 **Proceedings of an ACM conference on Language design for reliable software**, Volume 12 , 2 , 11 Issue 3 , 2 , 2Full text available:  pdf(539.58 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A within-subjects experimental design was used to test the effect of two variables on program understanding. The independent variables were complexity of control flow and paragraphing of the source code. Understanding was measured by having the subjects memorize the code for a fixed time and reconstruct the code verbatim. Also some subjects were asked to describe the function of the program after completing their reconstruction. The two groups of subjects for the experiment were students fr ...

Keywords: Paragraphing, Psychological complexity, Software psychology, Structured programming, Within-subjects

8 Design of self-checking software

S. S. Yau, R. C. Cheung

April 1975 **ACM SIGPLAN Notices , Proceedings of the international conference on Reliable software**, Volume 10 Issue 6Full text available:  pdf(688.95 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper discusses different techniques for constructing a piece of self-checking software for systems where ultra-reliability is required. Self-checking software can be designed to detect software errors, to locate and to stop the propagation of software errors, to assist in the recovery from errors and to verify the integrity of the system. Self-checking techniques can be implemented in the program to check the function, the control sequence and the data of a process. The functional asp ...

9 Interference between communicating parallel processes

Philip Gilbert, W. J. Chandler